Eine Standardlösung für die Integration elektrischer Geräte mit PROFINET IO und FDI

A Default Solution for Electrical Integration with PROFINET IO and FDI

Dr. rer. nat. **Dirk Schulz**, Dr.-Ing. **Jürgen Greifeneder**
ABB Corporate Research, Ladenburg

Kurzfassung

Bei der Integration elektrischer Geräte sind eine Reihe technologischer Entscheidungen zu treffen, wie z.B. ob PROFINET IO (Industrial Ethernet), PROFIBUS DP oder Modbus zur Anwendung kommt. Obwohl mit FDT DTMs und EDDL zwei häufig genutzte Integrationstechniken zur Verfügung stehen, finden bei dem Engineering elektrischer Geräte und Motor Control Centern (MCC) meist eigenständige Tools Verwendung. Der neue *Field Device Integration* (FDI) Standard führt die Vorteile von EDDL und FDT optimal zusammen.

Für eine Verwendung von PROFINET IO zusammen mit FDI sprechen zwei grundlegende Aspekte: Zum einen können die entsprechenden Lösungen im Ergebnis mit bestehenden Lösungen nicht nur mithalten, sondern übertreffen diese in einigen Aspekten. Zum anderen können diese Lösungen ohne zusätzlichen Engineeringaufwand bereitgestellt werden. FDI soll dabei zur Bedienung komplexer elektrischer Geräte und MCCs verwendet werden.

Abstract

There are a variety of technology choices to make when implementing a solution for electrical integration. Along with PROFINET IO (Industrial Ethernet), PROFIBUS DP and Modbus variants are still the used communication technologies. FDT DTMs and EDDL are available integration technologies and are often used to integrate electrical devices into the DCS, but it is not untypical to use stand-alone tools to engineer drives or motor control centers (MCC). With Field Device Integration (FDI), a new standard is appearing which combines the advantages of EDDL and FDT in a robust manner.

Introducing these most current technologies, PROFINET IO and FDI, is therefore considered from two angles: first of all, it is a *technology migration* that should provide a solution that is at least on par with the existing ones; secondly, it is an opportunity to make use of *new features* of the new technologies without additional work in engineering and operation. In any case, this means we expect FDI to handle not only simple instruments, but also complex electrical devices (including those than can be programmed similar to a small PLC or entire motor control centers).

1. Expectations on Electrical Integration

The core aspect of Electrical Integration arguably is to control electrical devices from an IEC61131 application via cyclic communication. This aspect is already well-established in the process industry, but reducing electrical integration to cyclic communication falls too short.

The complexity of electrical devices is particularly reflected in their parameterization process and the entirety of diagnostic data and alarms for devices, equipment, and process. PROFIBUS DP only provides a basic alarming mechanism that cannot compete with the level of details that PROFINET IO [6] devices can send in event-based manner. Also, PROFIBUS GSD or PROFINET GSDML [5] device description formats are suitable only for basic parameterization tasks, cannot describe process-related alarms, and focuses mostly on aspects of cyclic communication. Lastly, electrical devices and subsystems (motor controllers, drives, low-voltage switchgears, etc.) may come with a higher IO load than process instruments, but they benefit from a number of application profiles such as PROFIDrive, PROFIEnergy, safe communication via PROFISafe, low-voltage switchgear profiles, or fault descriptions according to DriveCom [4].

On the face of it, Industrial Ethernet protocols such as PROFINET IO (IEC 61158, IEC 61784) only add higher bandwidth and network redundancy, but as we will see in the remainder of this article, it has additional selling points with regard to integrating complex electrical devices; most importantly, all relevant profiles mentioned above are also available on top of PROFINET IO. Migrating communication to PROFINET IO therefore makes sense, provided we can complement GSD(ML) device description technology to handle the complex aspects of device engineering, alarms & events, and acyclic diagnosis. Our default solution uses Field Device Integration (FDI) technology to support all device management tasks as shown in Figure 1.

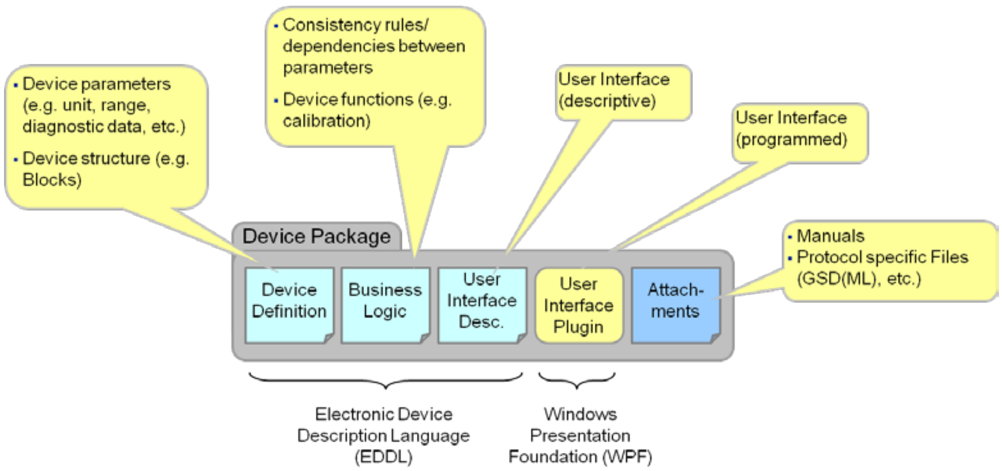


Figure 1 Providing all required device description artifacts in a single FDI Device Package; user interface plugin (UIP) and other attachments are (such as GSD) are optional by standard.

2. Solution Concept

2.1 Stakeholders of Electrical Integration

During plant operation, we consider the following information receivers for electrical devices

- Operator (operation/observation via electrical faceplates)
- Operator (alerting via process alarms in the operator alarm list)
- Maintenance Engineer (alerting via device and equipment events in system event list)
- Maintenance Engineer (alerting from device condition monitoring and root cause analysis from detailed diagnostic data)
- IEC 61131 Control Application (open-loop control)

Supplying these receives with the required information at runtime is a main purpose of device engineering for electrical devices. While this is already possible with existing technologies, our intention is to check which improvements PROFINET IO and FDI can add in combination and what is generally required when migrating to these technologies.
An essential part of device integration is mapping the specifics of the various fieldbus services into the DCS in a harmonized manner. For PROFINET IO, we focus on the following services:

- Cyclic communication (IO data ASE[1]), see section 2.4
- Acyclic communication (parameter record ASE), see section 0
- Alarming (alarm ASE), see section 2.5

2.2 Architecture Overview

Integrating actual device parameters and control data is very straight-forward. The main challenge -- illustrated in Figure 2 -- is the forwarding of diagnostic data from the device to the right stakeholders. Generally, the building blocks in the presented architecture overview represent any stereotypical DCS and particularly the device and server/client sides are entirely technology independent (i.e. the picture would look extremely similar for a PROFIBUS DP and FDT DTM solution).

Diagnostic data refers to any information that describes the (potentially critical) condition of process, device, and equipment. E.g. current levels L1,L2,L3 may be part of the cyclic image and used within the IEC 61131 control application, but they are also accessible via acyclic communication so we can show them in the FDI diagnostic menu for the device; an overcurrent will then trigger a channel alarm, which we show in the system event list, and if the

---

[1] PROFINET IO beschreibt seine Dienste in sogenannten *Application Service Elements*, ASE.

closed-loop control in the device trips as a result of this overcurrent, a time-stamped process alarm is forwarded into the process alarm list that is in focus of the operator. All the different data paths and information receivers eventually tap into the same supervised current value. Essentially, almost every component of the DCS is forwarder or receiver of some kind of diagnostic data.

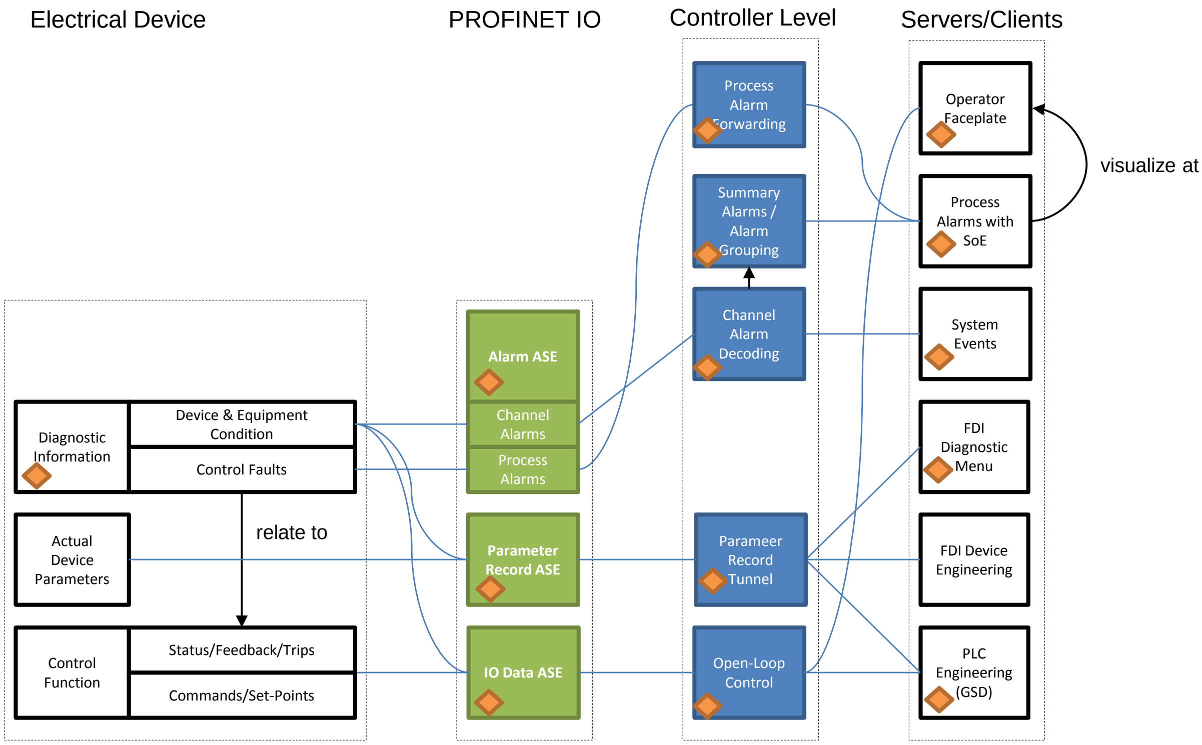


Figure 2 Flow of diagnostic information (faults, warning, status, etc.) via PROFINET IO

Our default approach provides devices, device descriptions, and system features in a manner that particularly this diagnostic information is made available in a meaningful way with a minimum amount of engineering. Part of this approach is to reduce the alarm load on the operator by creating summary alarms for the detailed events (logged in the system event list).

2.3 Note on Electrical Control Loops

Considering the control applications, there is a significant difference between instrumentation (sensing/actuation) and electrical control, as there are two nested loops, one between controller and device, one between device and electrical equipment (cf. Figure 3). Arguably, the latter is the more complex one: Even for torque or motion control with a drive, the PLC (upper loop) only provides a command or set-point and receives a corresponding feedback. In comparison, the amount of sensing and “actuation” (current modulation) that the drive (lower loop) has to perform is very extensive, taking a lot of load off the PLC and also allowing our electrical devices to run in stand-alone fashion (i.e. without DCS connection). In essence, this means that there is a device-internal distributed control application which must be configured and supervised.

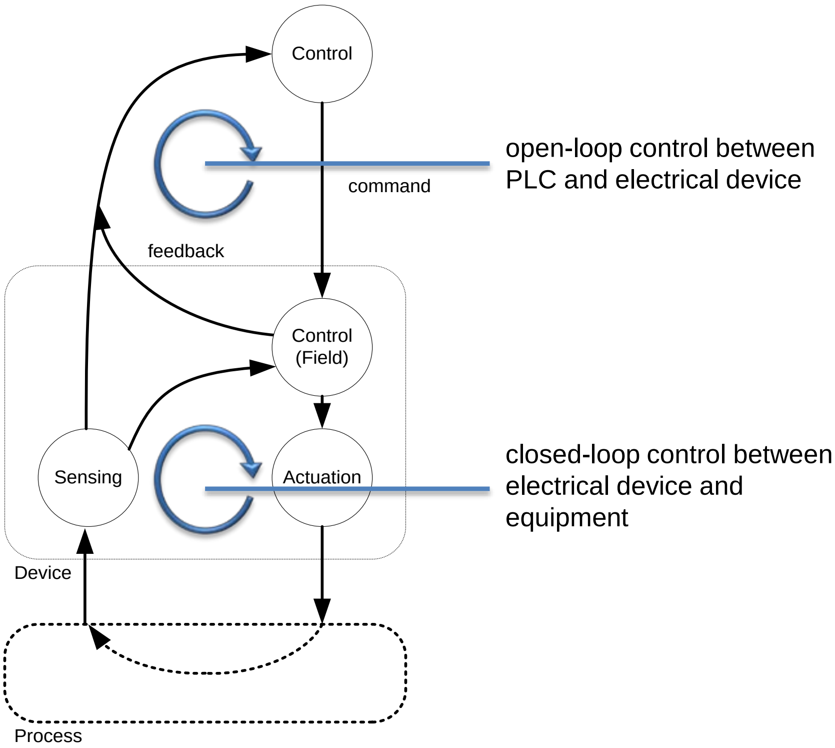


Figure 3 Nested Control Loops for Electrical Devices

2.4 PROFINET IO for Closed-Loop Control

Although the payload of cyclic communication does not change, PROFINET IO still offers significant benefit compare to e.g. PROFIBUS DP:

- **shared device (multi-master)**, which means several controllers can access the same device; while this may seldom occur for a single device (e.g. to monitor the feedback or trip from a device from another controller), the main use is when connecting entire subsystems like motor control centers (MCC); those may contain several dozens of motor starters, but not the entire MCC is necessarily controlled by the same PLC.
- **multiple connections to the same device**, which again is rather relevant for MCCs. Presuming an MCC with e.g. 60 motor starters, this leaves an odd 25 bytes per contained starter. This is typically sufficient, but with the option to tailor the IO data for each motor starter in a project, the desire to make additional measurement values (currents, temperatures) available to the IEC 61131 application also grows.
- **IO controller redundancy**, allowing a device to be simultaneously connected to two IO controllers (on the same or different PLCs).

- **communication redundancy**, which is one of the main features of Industrial Ethernet and allows to increase the reliability of communication beyond what a fieldbus cable can offer; from the PROFINET IO standard, we follow the recommendations to support RSTP (rapid spanning tree protocol) and MRP (media redundancy protocol).

Furthermore, existing features such as safe communication via PROFISafe are supported.

2.5 PROFINET IO for Alarms and Events

PROFINET IO specifies a powerful but complex alarming service, with a variety of alarm types, alarm specifiers, identifiers pointing to a PROFINET GSD device description, etc. Considering the information receivers listed in 2.1, we find that a small subset of the alarm service already fulfills our needs. In our default solution we use the following:

- Mandatory standard **system alarms** of PROFINET IO to indicate **communication-related** events (including the pulling/plugging of device modules).
- **Channel alarm** type to indicate faults and warnings of the electrical **device** (physical or firmware) based on the DriveCom profile (please refer to [1] for details on the diagnostic profile). Time-stamping occurs in the PLC.
- **Process alarms** with SoE (sequence of events) profile for faults of the **closed-loop control** between electrical device and equipment (see Figure 3). Providing a high-resolution time-stamp from within the device (not the PLC) is essential for SoE.

An advantage of PROFINET IO is the way that electrical devices cache alarm information: for each channel alarm, there is a defined index in the parameter record service that allows reading the alarm status acyclically; this means we align alarm lists for operation with diagnostic information in device management by default.

2.6 PROFINET IO for Acyclic Communication

PROFINET IO specifies a single communication service to access acyclic data (called parameter access service). This service is used to download parameters during communication start-up (like DPV0 in PROFIBUS DP). The same service is also used to access any acyclic data from (FDI) device management (like DPV1 and DVP2). This means our DCS can perfectly synchronize parameter values between PLC and device engineering for any given device without beforehand knowledge. Already mentioned (previous section 2.5) is the fact that also relevant alarm status information is available via this acyclic communication service.

2.7 FDI Device Management

Device Management includes several engineering and operational tasks for electrical devices:

- *During Engineering* and Commissioning of the device-internal functions (see Figure 5)
  - Basic Parameterization (units, ranges, modes of control, monitoring functions, etc.)
  - Logic Editing, connecting device-specific and general-purpose IO by logic functions running on the device
  - Start-Up Curves (for drives or soft-starters)
  - Advanced Parameterization (e.g. drive tuning parameters)
  - Subsystem Engineering (configuration of all subsystem properties, including hardware aspects for construction and ordering)

- During plant *operation*, the devices are monitored/supervised (see Figure 7 and Figure 8)
  - Condition Monitoring
  - Diagnosis (on demand)
  - Maintenance (device replacement, which is out of scope in this article)

Generally, the entire tool support may be implemented within FDI Device Package. Our proposed default solution comes very close to this as shown in Table 1.

#	**Task**	**FDI EDD**	**PROFINET GSD**	**FDI UIP**[2]	**Stand-Alone Tools**
1.1	basic parameterization	yes	yes		
1.2	logic editing			yes	
1.3	start-up curves	yes			
1.4	advanced parameterization	default			allowed
1.5	Subsystem Engineering	consider			yes
2.1	condition monitoring	yes[3]			
2.2	diagnosis	yes[4]			
2.3	maintenance (replacement)	default			allowed

Table 1 Covering Device Integration with PROFINET GSD and FDI Technology

---

[2] User Interface Plugin, essentially a software module written in Microsoft .NET.

[3] We expose the device health through a device condition variable according to NE 107 as defined by the FDI standard [2] part 5; for our default solution, we propose an extended health & service profile [8].

[4] There is a dedicated diagnostic root menu as part of an FDI EDD file.

3. Validation of FDI Technology for Complex Electrical Devices

3.1 FDI Device Package for Universal Motor Controller

The FDI standard represents a very comprehensive approach to modeling field devices; it may initially appear to be conceived mainly for instrumentation, but it defines features that aim at the support of remote IOs or other kind of subsystems. The FDI demonstrator presented at the Hannover Fair 2012 (cf. Figure 4) already included a basic integration of the Universal Motor Controller (UMC100) from ABB Stotz Kontakt.

Figure 4 FDI Demonstrator Setup for NAMUR General Assembly and Hannover Fair

To complete the FDI validation, we implemented a complete technology prototype for the UMC100, based on the PROFIBUS DP connectivity.[5] The FDI demonstration package includes both an EDD file and a UIP containing a logic editor for the device. These artifacts essentially support all device management use-cases as listed in Table 1.

Also considering the development effort when going from FDT DTMs to EDDL, we decided to generate the EDD file out of the description files of the current DTM product. Figure 5 shows a comparison of the original DTM (top) and the generated UIP skeleton (bottom). Already the auto-generated result comes very close, and remaining differences are mostly a result of the prototype state of the FDI host system. In any case, the generation approach provides a user interface description (UID) with no additional effort, and the resulting EDD file can of course be modified manually.

[5] For the purpose of evaluating FDI, i.e. acyclic data access, we find PROFIBUS DP and PROFINET IO to be sufficiently similar.

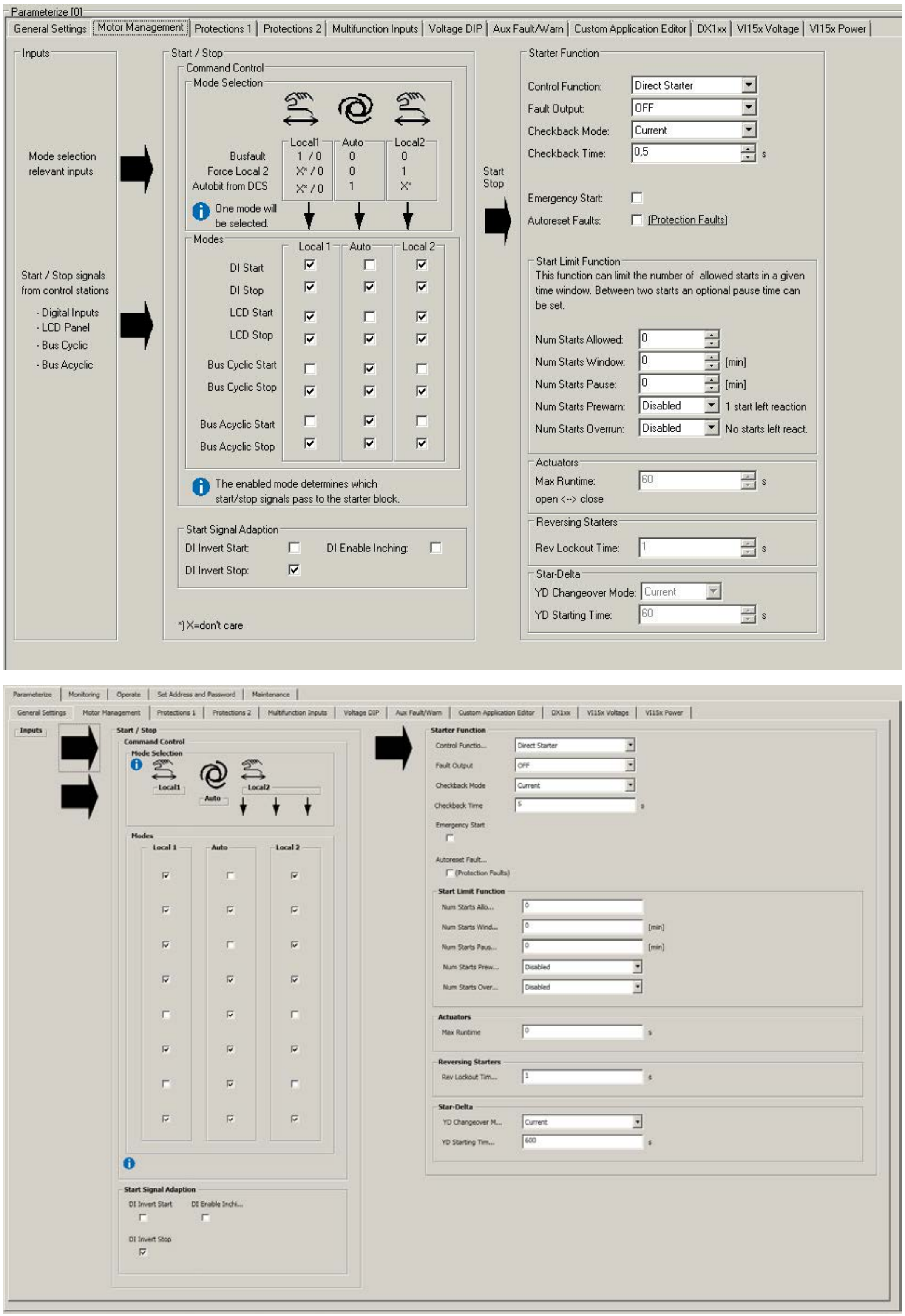


Figure 5 Side-By-Side Comparison of original FDT DTM (top) and auto-generated parts of the FDI User Interface Description (UID), which is described in EDDL (bottom).Showing the engineering for motor management.

We also integrated the existing logic editor (see Figure 6) linked from within the EDD menu.

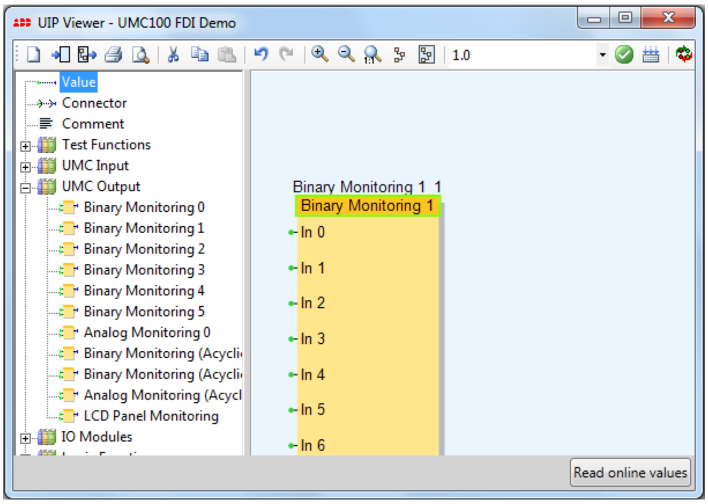


Figure 6 UMC100 Logic Editor, implemented with FDI User Interface Plugin technology

Furthermore, we integrated condition monitoring functionality into the EDD. FDI defines how to expose the device condition in the OPC UA [3] information model, following NAMUR recommendation NE107. In addition to this single variable, we expose more detailed conditions, root causes, and suggested actions in the information model and UID as shown in Figure 7. We find this level of detail to be helpful particularly when integrating device management systems with ERP systems for maintenance management (CMMS) [7].

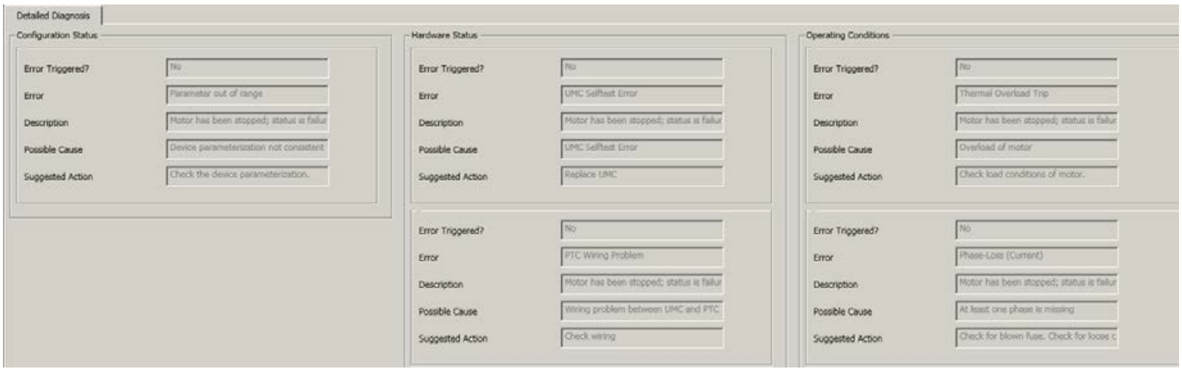
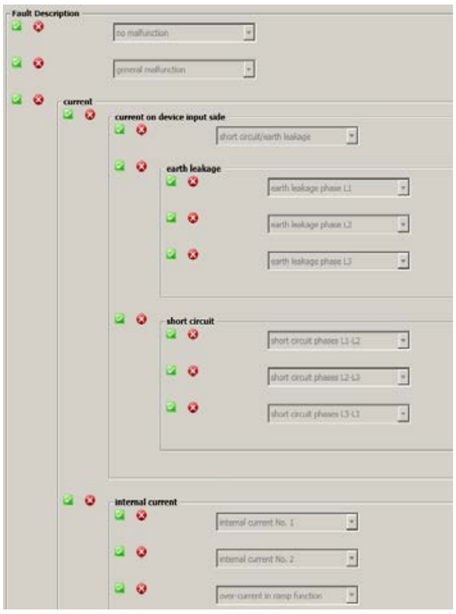


Figure 7 Visualization of Device Condition (from device-specific monitoring method in EDD)

Lastly we attempted a basic visualization of a *default condition profile for electrical devices* (please refer to our related article in [1]). This visualization is available from within the diagnostic root menu of the EDD, and it reflects exactly the events provided via the PROFINET IO channel alarms into the DCS A&E system (see previous section 2.5). While the initial rendering of the faults as shown in Figure 8 can certainly be improved (and maybe FDI

UIDs are not flexible enough), the main goal to show the alignment of alarms and events with device diagnostics was achieved. Again, we generated this part of the UID from a reference list of fault conditions also quoted in the figure.



no malfunction				0
general malfunction				1000
current				2000
	current on devic			2100
		short circuit/earth leakage		2110
		earth leakage		2120
			earth leakage phase L1	2121
			earth leakage phase L2	2122
			earth leakage phase L3	2123
		short circuit		2130
			short circuit phases L1-L2	2131
			short circuit phases L2-L3	2132
			short circuit phases L3-L1	2133
	internal current			2200
			internal current No. 1	2211
			internal current No. 2	2212
			over-current in ramp function	2213

Figure 8 Generic "Default Condition Profile" for Electrical Devices, generated from DriveCom Faults table

3.2 Conclusion on FDI Technology

The core of device integration artifacts are the PROFINET GSD and EDD files. They cover 80% of all typical use-cases for electrical devices. There are few exceptions where we require the use of binary components such as FDI User Interface Plugins (UIP) or stand-alone tools; these cases concern editing of project-specific control logic for a device and engineering entire electrical subsystems.

This means that, unlike for instrumentation, there is a clear need to have binary tools to handle some aspects of complex devices and subsystems. With FDI, the integration of these components into the device management systems is actually the preferred choice. With EDD, this was not possible and with FDT we have known versioning issues with the installed DTMs. With FDI, the binary components are delivered from server to client on demand, i.e. no installation is necessary on the client, and different device versions/revisions can be supported simultaneously; thanks to the Microsoft .NET runtime, EDD methods and UIPs are executed in a sandboxed environment (much like a virtual machine), so the DCS stability is never at risk.

FDI thus fills a gap between EDD and FDT DTM technology to provide a seamless and stable integration of all device aspects into the DCS. Stand-alone tools may still be used, but are technologically not required.

4. Conclusion – Benefit for Customers and Automation Vendors

There is a number of technical benefits when migrating from PROFINET IO and classic EDD or FDT DTMs to PROFINET IO and FDI technology. Besides the evident advantages of Industrial Ethernet like higher bandwidth and communication redundancy, the aligned protocol services for alarming and acyclic data of PROFINET IO allow a better alignment of alarming and device management in the DCS. For device integration, the solution complements established device description formats such as GSDML and EDDL with the robustness of modern software technology. Using FDI allows us to implement the full range from simple to very complex device management tasks on top of a single integration standard.

For the customers, this results in a more seamless integration of electrical devices between operation and device (asset) management from a single device package. Process-critical faults are logged as events with high-precision time stamps to allow building a sequence of events following a fault. By basing diagnostic data on a common default profile for all electrical devices [1], we align alarms & events and acyclic diagnostic information in FDI perfectly. The concurrent use of multiple device package versions is a prerequisite for care-free device replacement and updates. The automatic deployment and sandboxed execution of these packages greatly improves the usability of the device management system.

For a universal motor controller as a complex, programmable device we have shown the feasibility to use FDI as device integration technology to complement the PROFINET GSD. This approach allows handling complex electrical devices, using only open standard technologies. What remains in order to finish the lab validation work for FDI from our perspective is the integration of a modular subsystem like an entire motor control center.

References

[1] Greifender, Jürgen; Schulz, Dirk: Profile based IEC 61131 Function Libraries for Efficient Electrical Engineering, Automation 2013, Baden Baden, 2013.

[2] IEC 62769, Field Device Integration (FDI)

[3] OPC UA For Devices 1.00 Companion Specification, Dec 19, 2009

[4] DRIVECOM Nutzergruppe e.V.: DriveServer Spezifikation V 1.1, http://www.drivecom.org

[5] GSDML (GSD Markup Language), Technische Spezifikation von PROFINET IO version 2.3, PNO Order No.: 2352, 2011.

[6] IEC 61784 and IEC 61158 [family of standards including PROFINET IO]

[7] Schulz, Dirk; Gitzel, Ralf: Seamless Maintenance - Integration of FDI Device Management & CMMS, Automation 2013, Baden Baden, 2013.